

Movie Recommendation System

Kuldeep Kumar Singh, Priyanshu Bora, Arman Grover, Sajal Singh Masand, Yadender Singh

Department of Computer Science and Engineering

Chandigarh University Mohali, Punjab

Abstract- In the era of information overload, it is very difficult for users to get information that they are really interested in. The mission of Recommendation System is to connect users and information, which in one way helps users to find information valuable to them and in another way push the information to specific users. The need for the hour is to develop some code that can tell at a beginner level the matching pattern of the customer trend and recommend him the best item of his interest level. This will help us in making the customer experience satisfactory and able to achieve good ratings and popularity as well.

Keywords- Movie Recommendation, Prediction, Logical Database, Deep Learning, Machine Learning, Phyton API, Super Computers, Recommendation Engine, Collaborative Filtering.

I. INTRODUCTION

1.1 Contemporary Issue In the era of information overload, it is very difficult for users to get information that they are really interested in. The mission of Recommendation System is to connect users and information, which in one way helps users to find information valuable to them and in another way push the information to specific users.

1.2 Identification of Problem -The need for the hour is to develop some code which can tell at a beginner level the matching pattern of the customer trend and recommend him with the best item of his interest level. This will help us in making the customer experience satisfactory.

1.3 Identification of Task -For building a recommender system from scratch, we deal with several tasks including:

1. Product sourcing- This involves finding and selecting movies and web series to list on the system. This can include researching suppliers, negotiating prices, and determining product demand.

2. Design and Development- This involves creating a recommendation system that is user-friendly, visually appealing, and optimized for search engines.

3. Content creation- This involves creating highquality movie descriptions, images, and videos that will entice customers to make a selection.

4. Order processing- This involves receiving and processing customer requests, managing inventory, and fulfilling all requests.

5. Data analysis- This involves monitoring and analysing system traffic, sales data, and user behaviour to make informed business decisions and improve the user experience.

So, we get these questions

1. How to recommend movies when there is ample user information.
2. What kind of movie features can be used for the recommender system.
3. How to calculate the similarity between two movies.

4. How to enhance viewing experience and customer satisfaction.

1.3.3 Implementation Plan Recommender system has developed for many years, whichever entered a low point. In the past few years, the development of machine learning, largescale network and high-performance computing is promoting new development in this field. We will consider the following aspects in future work.

1. Use collaborative filtering recommendation. After getting enough user data, collaborative filtering recommendation will be introduced, collaborative filtering is based on the social information of users, which will be analysed in the future research.

2. Introduce more precise and proper features of movie. Typical collaborative filtering recommendation use the rating instead of object features. In the future we should extract features such as colour and subtitle from movie which can provide a more accurate description for movie.

3. Introduce user dislike movie list. The user data is always useful in recommender systems. In the future we will collect more user data and add user dislike movie list. We will input dislike movie list into the recommender system as well and generate scores that will be added to previous result.

4. Introduce machine learning. For future study, dynamic parameters will be introduced into recommender system, we will use machine learning to adjust the weight of each feature automatically and find the most suitable weights.

II. ORGANIZATION OF THE REPORT

1. Executive Summary- This section provides a brief overview of the key findings and recommendations of the report. It should highlight the most important points and give readers a sense of what they can expect to find in the report.

2. Introduction- The introduction should provide background information about the recommendation platform, its products or services, and its target audience. It should also outline the purpose and scope of the report.

3. System Analysis- This section should analyse the system's design, usability, content, and functionality. It should identify any strengths and weaknesses and provide recommendations for improvement.

4. Customer Analysis- This section should provide insights into the platform's customers, including their demographics, behaviour, and preferences. It should also include feedback from customer surveys or reviews.

5. Recommendations- This section should provide actionable recommendations for improving the platform's performance. It should prioritize the most important recommendations and provide specific steps for implementation.

6. Conclusion- The conclusion should summarize the key findings and recommendations of the report. It should also provide insights into the potential impact of implementing the recommendations.

7. References- This section should include a list of sources cited in the report, including academic articles, industry reports, and online resources.

8. Languages used- Python, Machine Learning.

even when the viewer had previously watched and rated films with female leads.

- 2019: The film "The Social Dilemma" drew attention to the ways in which social media algorithms can be used to manipulate user behaviour and reinforce existing biases.
- 2020: With the COVID-19 pandemic forcing more people to stay at home and rely on streaming services for entertainment, concerns about the potential negative effects of recommendation algorithms grew. In August, the documentary film "The Social Dilemma" was released on Netflix, further raising awareness about these issues.
- 2021: Studies continue to investigate the biases and limitations of recommendation algorithms, as well as potential solutions for addressing these issues. For example, some researchers have suggested incorporating user diversity and feedback into the algorithm to reduce bias, while others have proposed using more transparent algorithms that are easier for users to understand and critique.

2. Existing Solutions Movie recommendation systems are designed to provide personalized recommendations for movies to users based on their viewing history and preferences. There are several types of movie recommendation systems, including content-based, collaborative filtering, and hybrid recommendation systems. In this answer, I will provide a detailed description of each of these solutions.

III.LITERATURE REVIEW

BACKGROUND STUDY

1. Timeline of the reported problem

- 1990s: The concept of personalized movie recommendation systems first emerged, with early examples including systems developed by Group Lens Research and Firefly Networks.
- 2000s: As online streaming services like Netflix and Amazon Prime Video grew in popularity, the use of recommendation algorithms became more widespread. Netflix even launched the Netflix Prize competition in 2006, offering a \$1 million prize to anyone who could improve their recommendation algorithm by 10%.
- 2010s: Concerns began to arise about the potential biases and limitations of recommendation algorithms. In 2011, researchers at MIT found that Netflix's recommendation algorithm tended to reinforce gender and racial stereotypes in its suggestions.
- 2016: The documentary film "The Great Hack" explored the ways in which data analysis and recommendation algorithms can be used to manipulate public opinion and influence elections. This raised concerns about the potential dangers of these algorithms when used on a large scale.
- 2018: A study by researchers at the University of Michigan found that recommendation algorithms used by services like Netflix and Amazon Prime Video tended to recommend films with male-dominated casts,

3. Content-Based Recommendation System- Content-based recommendation systems suggest movies to users based on their preferences and the similarity of the movie content. The system uses the metadata of the movie (actors, directors, genres, plot, etc.) to identify similarities between movies. For example, if a user watches a lot of action movies, the system will recommend other action movies that have similar plots, actors, or directors. Pros: Content-based recommendation systems are easy to implement, and they can provide accurate recommendations for users who have a specific genre or preference. Cons: Content-based recommendation systems are limited to the metadata available, which may not capture all the nuances of a movie, and they may not be able to suggest movies outside of the user's preferred genre. Figure 1: Content-Based & Collaborative Filtering.

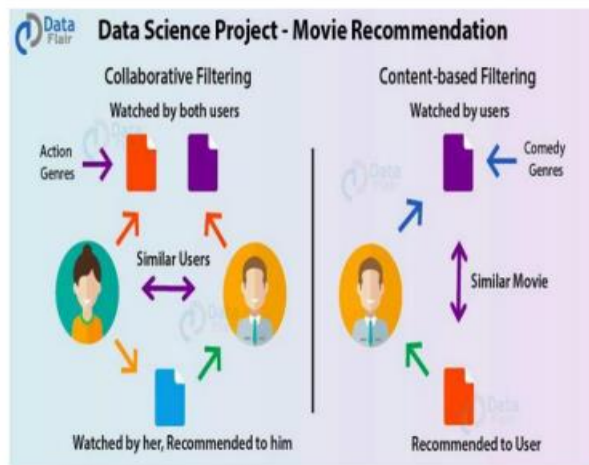


Fig.1 Credit: Data Flair.

2. Collaborative Filtering Recommendation System:

Collaborative filtering recommendation systems are based on the idea that people who share similar interests will like similar movies. The system uses data from other users to suggest movies to a particular user. For example, if a user likes movies that other users with similar viewing history like, the system will suggest movies that those users enjoyed. Pros: Collaborative filtering recommendation systems can provide personalized recommendations for users, and they can suggest movies outside of a user's preferred genre. Cons: Collaborative filtering recommendation systems require a large amount of data to be effective, and they may not provide accurate recommendations for users who have unique preferences.

3. Hybrid Recommendation System:-

Hybrid recommendation systems combine content-based and collaborative filtering recommendation systems to provide more accurate recommendations. The system uses both the user's viewing history and the metadata of the movie to suggest movies. For example, if a user likes movies with a particular actor, the system will suggest movies with that actor, but also considers other users' viewing history to make recommendations. Pros: Hybrid recommendation systems provide more accurate recommendations than content-based or collaborative filtering recommendation systems alone. Cons: Hybrid recommendation systems can be more complex and difficult to implement than other recommendation systems. Figure 2. Recommendation Example Credit: Data Flair In conclusion, movie recommendation systems use different algorithms to provide personalized recommendations for users. Each algorithm has its advantages and disadvantages, and it is important to consider the type of data available and the preferences of the users to choose the most effective recommendation system.

3. Bibliometric Analysis- 1. "A Collaborative Filtering Algorithm and Its Application to Personalized Movie Recommendations" by Breese et al. (1998): This paper

introduced collaborative filtering as a technique for movie recommendations. Collaborative filtering works by analysing user behaviour and making recommendations based on the behaviour of similar users. The authors proposed a method for building a user-based collaborative filtering algorithm and showed that collaborative filtering can provide more accurate recommendations than content-based approaches.

4. "Context-aware movie recommendation system-" by S. Gupta et al. (2016) proposed a movie recommendation system that considers the context of the user's current situation, such as location and time of day, to provide more personalized recommendations.

5. "Deep learning-based movie recommendation system" by X. Li et al. (2015) proposed a movie recommendation system that uses deep learning techniques to learn the features of movies and users to make better recommendations. The study showed that this method outperformed traditional collaborative filtering techniques.

6. "Movie recommendations based on sentiment analysis of Twitter data" by T. Hossein et al. (2018) proposed a movie recommendation system that uses sentiment analysis of Twitter data to predict user preferences. The study showed that this method can improve the accuracy of movie recommendations.

7. "A hybrid movie recommendation system using content-based and collaborative filtering" by S. Zhang et al. (2012) proposed a movie recommendation system that combines content-based and collaborative filtering techniques to provide more accurate recommendations. The study showed that this method outperformed traditional collaborative filtering techniques.

8. "Factorization Meets the Neighbourhood- A Multifaceted Collaborative Filtering Model" by Koren (2008): This paper proposed a matrix factorization approach for movie recommendations that combines user-based and item-based collaborative filtering. The matrix factorization approach works by factorizing the user-item rating matrix into lower-dimensional matrices that capture latent features of users and items. The study showed that the proposed approach outperformed traditional collaborative filtering methods.

9. "Temporal Dynamics of Learning in a Movie Recommendation System" by Koren et al. (2009): This paper extended the matrix factorization approach proposed in the previous study by incorporating temporal dynamics into the model. The authors showed that users' preferences can change over time and that incorporating temporal dynamics into the model can improve the accuracy of movie recommendations. The study also introduced a new dataset for evaluating movie recommendation algorithms.

10. "Content-Based Recommendation Systems" by Pazzani and Billsus (2007): This paper proposed content-based filtering as an alternative approach to collaborative filtering for movie recommendations. Content-based filtering works by analysing the attributes of items and making recommendations based on items that are like those previously liked by the user. The authors showed that content-based filtering can be effective in providing recommendations for new users or items with little to no rating data.

11. "Combining Collaborative Filtering with Personal Agents for Better Recommendations" by Burke et al. (2005)- This paper proposed a hybrid recommendation system that combines collaborative filtering with personal agents that represent users' preferences and interests. The personal agents can provide additional information about users that cannot be captured by collaborative filtering alone. The authors showed that the hybrid approach can provide more personalized and diverse recommendations.

12. "Neural Collaborative Filtering" by He et al. (2017)- This paper proposed a deep learning-based approach for movie recommendations that combines matrix factorization with neural networks. The authors showed that the proposed approach outperformed traditional collaborative filtering methods and other deep learning-based approaches. The deep learning-based approach can capture complex nonlinear relationships between users and items and can provide more accurate recommendations.

13. "Sentiment Analysis of Movie Reviews for Recommendation" by Pang and Lee (2008)- This paper proposed a method for incorporating sentiment analysis of movie reviews into movie recommendations. The authors showed that the sentiment-based approach can improve the relevance and quality of recommendations. The sentiment analysis can capture users' emotional responses to movies and can provide more personalized recommendations. Overall, these studies show that movie recommendation systems can be improved through various techniques, such as collaborative filtering, content-based filtering, hybrid approaches, deep learning, sentiment analysis, and temporal dynamics. These techniques can provide more accurate, personalized, and diverse recommendations to users, and can ultimately improve the user experience of online streaming services.

III. REVIEW SUMMARY

1. Types of recommendation systems- There are primarily two types of movie recommendation systems - collaborative filtering and content-based filtering. Collaborative filtering suggests movies based on the viewing behaviour of other users with similar preferences, while content-based filtering recommends movies based on the similarity of movie attributes.

2. Hybrid recommendation systems- These systems combine both collaborative filtering and content-based filtering to provide more accurate recommendations to users.

3. Data pre-processing- To train a recommendation system, it is essential to pre-process the data by removing irrelevant data, handling missing data, and normalizing the data.

4. Evaluation metrics- Evaluation metrics such as precision, recall, F1 score, and mean absolute error are used to assess the performance of a recommendation system.

5. Cold start problem- A common challenge faced by recommendation systems is the cold start problem, which occurs when a new user or a new movie is added to the system. To address this issue, recommendation systems may use techniques such as content-based filtering, demographic-based filtering, and knowledge-based filtering.

6. Deep learning-based recommendation systems- Deep learning-based models such as neural networks, autoencoders, and deep belief networks have been used to develop more sophisticated recommendation systems.

7. Ethics and privacy- Recommendation systems should be designed to ensure user privacy and avoid bias or discrimination based on factors such as race, gender, and age.

8. Problem Definition -The problem definition of a movie recommendation system is to provide personalized movie recommendations to users based on their past viewing history and preferences. The primary goal of the movie recommendation system is to enhance the user's viewing experience by suggesting movies that they are likely to enjoy. The problem can be broken down into several subproblems, including data collection, data preprocessing, algorithm selection, and evaluation.

Data collection involves gathering data about the user's viewing history, which includes information such as the movies they have watched, the time they watched them, and the ratings they gave to each movie. Data preprocessing involves cleaning and transforming the raw data into a format that can be used by the recommendation system. This may include removing duplicates, filling in missing data, and normalizing the data. Algorithm selection involves choosing the most appropriate recommendation algorithm for the given data. The three main types of recommendation algorithms are content-based, collaborative filtering, and hybrid recommendation systems.

9. Goals/Objectives -The goal/objective of a project on a movie recommendation system is to create a software system that suggests movies to users based on their past viewing history and preferences. The system should improve the user experience by helping them discover

movies they are likely to enjoy and reducing the time spent searching for new movies. Additionally, the project can also aim to increase user engagement, retention, and revenue for the streaming platform by providing a personalized and satisfying viewing experience. To achieve these goals, the project may involve several steps, such as collecting and processing data on user behaviour, building machine learning models to generate recommendations, testing, and evaluating the system's performance, and integrating it into a user-friendly interface.

IV. DESIGN FLOW/ PROCESS 3.1 EVALUATION AND SELECTION OF SPECIFICATION/ FEATURES

Designing a movie recommendation system involves selecting and evaluating various specifications and features that will help to create a system that provides relevant movie recommendations to users. Some important considerations for selecting the specifications and features of a movie recommendation system:

- 1. Data sources-** The recommendation system requires a large dataset of movie ratings and user preferences. The system should be able to extract data from various sources such as movie databases, user reviews, and social media platforms.
- 2. Recommendation algorithms-** The system should employ various recommendation algorithms such as collaborative filtering, content-based filtering, and hybrid recommendation to generate personalized movie recommendations.
- 3. User interface-** The recommendation system should have an intuitive and user-friendly interface that provides recommendations in an easy-to-use and visually appealing format.
- 4. Personalization-** The system should be able to personalize recommendations based on user preferences, history, and behavior.
- 5. Accuracy-** The system should be able to provide accurate recommendations that match the user's preferences and interests.
- 6. Diversity-** The system should provide a diverse range of recommendations, so users can explore different genres, actors, and directors.
- 7. Feedback-** The system should allow users to provide feedback on recommendations, so the system can learn from their preferences and improve the accuracy of future recommendations.
- 8. Integration-** The recommendation system should be able to integrate with other platforms such as streaming services, so users can easily access recommended movies. By considering these factors, a movie recommendation system can be designed and developed that will provide accurate and personalized movie recommendations to users.

V. DESIGN CONSTRAINTS

When designing a movie recommendation system, there are several design constraints that need to be taken into consideration. These constraints may limit the design options available and may have a significant impact on the final product.

- 1. Data Availability-** The amount and quality of available data can limit the effectiveness of the recommendation system.
- 2. Technical Constraints-** The technical constraints such as processing power, storage capacity, and network bandwidth can limit the performance and scalability of the recommendation system.
- 3. Privacy and Security-** The recommendation system may need to comply with privacy laws and regulations, and it should protect users' data from unauthorized access and security breaches.
- 4. Time and Budget Constraints-** Time and budget constraints can impact the design, development, and deployment of the recommendation system. It may limit the number of features, quality of the user interface, or the scale of the system.
- 5. Integration-** The recommendation system may need to integrate with other platforms or systems. Integration constraints can limit the compatibility and interoperability of the recommendation system.
- 6. User Interface Constraints-** User interface constraints such as the device limitations, user preferences, and accessibility requirements can impact the design of the recommendation system. The design team can develop a recommendation system that meets the business requirements, user needs, and technical constraints by taking above constraints into consideration.

V. ANALYSIS OF FEATURES AND FINALISATION SUBJECT TO CONSTRAINTS

When analyzing the features of a movie recommendation system and finalizing them, it's essential to consider the design constraints that may impact the system's effectiveness and usability. The step-by-step approach to analyzing and finalizing the features of a movie recommendation system:

- **Identify the design constraints:** The first step is to identify the design constraints such as data availability, technical constraints, privacy and security, time and budget constraints, integration, user interface constraints, and localization.
- **Define the user requirements:** Define the user requirements based on the user's expectations, behavior, and preferences. This can be done through user surveys, interviews, and analysis of user data.
- **Evaluate the available data:** Evaluate the available data and determine if it's sufficient to generate accurate recommendations. If there is limited data, consider using

data augmentation techniques or partnering with other platforms to increase the available data.

- Evaluate recommendation algorithms: Evaluate different recommendation algorithms such as collaborative filtering, content-based filtering, and hybrid recommendation to determine which algorithms are most effective for the available data and user requirements.
- Develop a user interface: Develop a user interface that is intuitive and easy to use. Consider the user interface constraints such as device limitations, user preferences, and accessibility requirements.
- Personalization: Develop personalization features that match the user's preferences, history, and behavior.
- Privacy and security: Implement privacy and security measures that comply with privacy laws and regulations and protect user data from unauthorized access and security breaches.
- Finalize features subject to constraints: Finalize the features subject to design constraints such as technical constraints, time and budget constraints, and integration constraints. Identify the essential features that are critical to the system's success and prioritize them.

1. Design Flow - The design flow of a movie recommendation system involves several steps, from data collection and analysis to algorithm development and system deployment. Here is a typical design flow for a movie recommendation system.

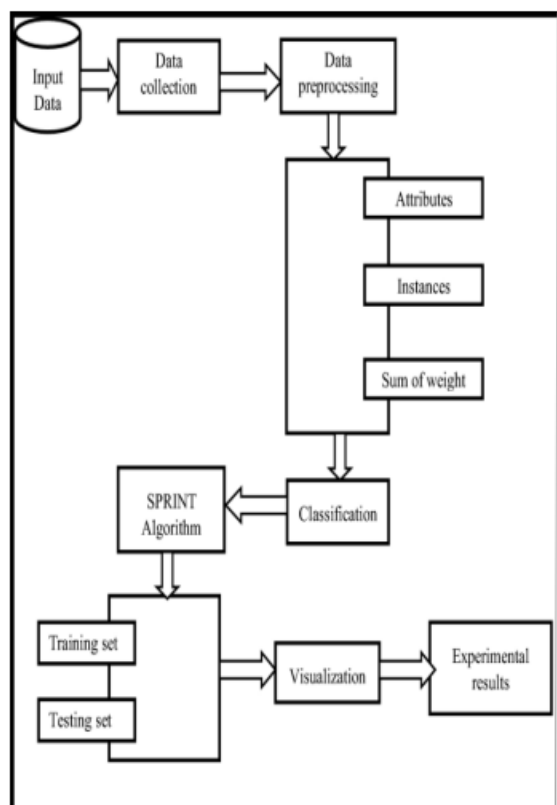


Fig.2 Credits: Data Flair 1.

1. Data collection- The first step in designing a movie recommendation system is to collect relevant data. This includes data about movies, such as their title, genre, actors, director, release date, and rating, as well as user data, such as their viewing history, ratings, and reviews. The data can be collected from various sources, including movie databases, streaming platforms, and social media.

2. Data preprocessin:- Once the data has been collected, it needs to be preprocessed. This involves cleaning and transforming the data to make it suitable for analysis. For example, the data may need to be standardized or normalized to remove any outliers or inconsistencies. **3. Data analysis:** The preprocessed data is then analyzed to identify patterns and relationships between movies and users. This can be done using various techniques, including data mining, clustering, and classification. The goal is to identify similarities between movies and users, which can be used to make recommendations.

4. Algorithm development- Based on the results of the data analysis, an algorithm is developed to generate recommendations. This algorithm can be based on various approaches, including collaborative filtering, content-based filtering, and hybrid recommendation. The algorithm takes into account the user's viewing history, preferences, and behavior, as well as the characteristics of the movies.

5. Model training- Once the algorithm has been developed, it needs to be trained using the preprocessed data. This involves using machine learning techniques to optimize the algorithm based on the available data. The goal is to ensure that the algorithm generates accurate and relevant recommendations.

6. System testing- Once the system is developed, it needs to be tested to ensure that it is functioning as expected. This involves testing the system for accuracy, performance, and usability. The system should be tested using both simulated and real-world scenarios.

7. Deployment- Finally, the system is deployed to production, and it becomes available to users. Ongoing monitoring and maintenance are necessary to ensure that the system continues to provide accurate and relevant recommendations.

VI. IMPLEMENTATION PLAN METHODOLOGY

The implementation plan of a movie recommendation system involves several steps, from development to deployment. Here's a typical implementation plan for a movie recommendation system:

1. System architecture design- The first step in the implementation plan is to design the system architecture. This includes designing the data model, user interface, and algorithmic components.

2. Problem definition- The next step is to define the problem that the movie recommendation system will solve. This involves identifying the target audience, the data sources, and the desired outcomes.

3. Data collection and preprocessing- The next step is to collect and preprocess the data. This involves cleaning and transforming the data to make it suitable for analysis. The data can be collected from various sources, including movie databases, streaming platforms, and social media.

4. Model training and validation- Once the algorithm has been developed, it needs to be trained using the preprocessed data. This involves using machine learning techniques to optimize the algorithm based on the available data.

5. System integration and testing- The recommendation algorithm is integrated into the overall system architecture, which includes the user interface and database. The system is then tested to ensure that it is functioning as expected.

6. Deployment and maintenance- Finally, the system is deployed to production, and it becomes available to users. Ongoing monitoring and maintenance are necessary to ensure that the system continues to provide accurate and relevant recommendations.

7. Uploading Dataset

```
In [1]: import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd

In [2]: movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')

In [3]: movies.head()
```

Fig. 3. Uploading dataset.

8. Refining of dataset

```
In [7]: movies.dropna(inplace=True)

In [8]: movies.duplicated().sum()
Out[8]: 0

In [9]: movies.iloc[0].genres
Out[9]: '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]'

In [10]: import ast

In [11]: def convert(obj):
    L = []
    for i in ast.literal_eval(obj):
        L.append(i['name'])
    return L

In [12]: movies['genres'] = movies['genres'].apply(convert)

In [13]: movies.head()
```

3. Fetching Data values

In summary, the methodology for a movie recommendation system involves designing the system architecture, collecting and preprocessing data, training and testing the system, deploying and maintaining it over time. By following these, it's possible to create a movie recommendation system that provides accurate and personalized recommendations to users.

VII. RESULT ANALYSIS AND VALIDATION

1. Implementation Of Solution

The implementation of a solution for a movie recommendation system involves the process of translating the design into a working system that can generate accurate and relevant recommendations to users. Training Data Model:

```
In [30]: import nltk

In [31]: from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()

In [32]: def stem(text):
    y=[]
    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)

In [33]: new_df['tags']=new_df['tags'].apply(stem)

In [34]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 5000 , stop_words = 'english')

In [35]: vectors=cv.fit_transform(new_df['tags']).toarray()
```

Figure 4 Importing libraries.

Code: import pickle import streamlit as st import requests
def fetch_poster(movie_id): url = "https://api.themoviedb.org/3/movie/{ }?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US".format(movie_id) data = requests.get(url) data = data.json() poster_path = data['poster_path'] full_path = "https://image.tmdb.org/t/p/w500/" + poster_path return full_path
def recommend(movie): index = movies[movies['title'] == movie].index[0] distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1]) recommended_movie_names = [] recommended_movie_posters = [] for i in distances[1:6]: # fetch the movie poster movie_id = movies.iloc[i[0]].movie_id

```
recommended_movie_posters.append(fetch_poster(
movie_id))
recommended_movie_names.append(movies.iloc[i[
0]].title)
return
recommended_movie_names,recommended_movie
_posters
st.header('Movie Recommender System')
movies = pickle.load(open('movie_list.pkl','rb'))
similarity = pickle.load(open('similarity.pkl','rb'))
movie_list = movies['title'].values
selected_movie = st.selectbox( "Type or select a movie from the dropdown", movie_list )
if st.button('Show Recommendation'):
recommended_movie_names,recommended_movie
_posters = recommend(selected_movie)
col1, col2, col3, col4, col5 = st.beta_columns(5)
with col1:
st.text(recommended_movie_names[0])
with col2:
st.image(recommended_movie_posters[0])
with col3:
st.text(recommended_movie_names[1])
with col4:
st.image(recommended_movie_posters[1])
with col5:
st.text(recommended_movie_names[2])
with col1:
st.image(recommended_movie_posters[2])
with col2:
st.text(recommended_movie_names[3])
with col3:
st.image(recommended_movie_posters[3])
with col4:
st.text(recommended_movie_names[4])
with col5:
st.image(recommended_movie_posters[4])
```

4.2 RESULT/OUTPUT After typing the following command to execute the information from dataset, you will be redirected to website on chrome which will run your streamlit app. Figure 4.6: Output Screen Figure 4.7: Output Screen showing recommendation.

4.2 Result/Output



Figure 5 Output Screen.

After typing the following command to execute the information from dataset, you will be redirected to website on chrome which will run your streamlit app.

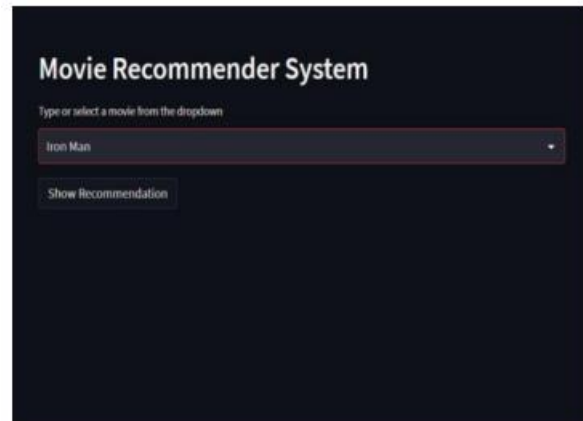


Figure 6 Output Screen.



Figure 7 Output Screen showing recommendation.

After clicking on “Show Recommendation”, it will recommend you few movies that you might be interested in.

VII.CONCLUSION AND FUTURE WORK

Conclusion Recommendation system is a data filtering technique. They are used to provide suggestions to the users according to their interest and need. It is a very popular technique in recent years and used by many e-commercial websites and other platforms to recommend news, books, movies, shopping items, novels, music and much more that is why recommendation system has become a hot topic. Techniques like Clustering, Similarity and Classification are used to get better. In future we can work on hybrid recommender using clustering and similarity for better performance. Our approach can be further extended to other domains to recommend songs, video, venue, news, books, tourism and ecommerce sites, etc. Recommendation systems are very useful and effective technique of filtering the data. This paper is a review of recommendation systems that will describe

recommendation system, how it works and helps in different platforms, and the different types of it with their merits and limitations.

5.2 Future Work Here are some potential areas for future work in movie recommendation systems:

1. Personalization: Current recommendation systems often treat all users the same, providing the same set of recommendations to everyone. In the future, recommendation systems may become more personalized, taking into account individual user preferences, demographics, and behavior patterns.
2. Use of more data sources: In the future, recommendation systems may incorporate data from a wider range of sources, including social media, search history, and user-generated content. This could lead to more accurate and diverse recommendations.
3. Multi-modal recommendation: Current recommendation systems primarily rely on text data to generate recommendations, such as movie titles, descriptions, and reviews. In the future, recommendation systems may incorporate other forms of data, such as images, audio, and video, to generate more diverse and engaging recommendations.
4. Integration with emerging technologies: With the rapid advancement of technologies such as augmented reality (AR), virtual reality (VR), and artificial intelligence (AI), there is an opportunity to integrate movie recommendation systems with these technologies. For example, an AR-based movie recommendation system could provide personalized movie recommendations based on the user's current surroundings.

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project review report work. We would like to show our greatest appreciation to Ms. Shalini Singh, Project Supervisor at Chandigarh University, Mohali, Punjab and Ayush Rai, 3rd year student at Chandigarh University. We always feel motivated and encouraged every time by their valuable advice and constant inspiration; without their encouragement and guidance this project would not have materialized.

Also, the support of various research paper written by dignified writers and authors helped in collecting the facts and figures for the review report. Apart from that Wikipedia gracifies our search for content to grasp and go through various key aspects of movie recommendation procedures. Collaborative work of our team helped in successfully deriving this research paper for reference to our academic grades and project. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project research paper.

REFERENCES

- [1]. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Data Engineering*, 17(6), 734-749.
- [2]. Bobadilla, J., Ortega, F., Hernando, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109-132.
- [3]. Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 43-52.
- [4]. Goldberg, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.
- [5]. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions & Informations Systems*, 22(1), 5-54.
- [6]. Koren, Y. (2008). Factorization meets the neighbourhood: A multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426-434.
- [7]. Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76-80.
- [8]. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web*, 285-295.
- [9]. Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender Systems Handbook* (pp. 257-297). Springer US.
- [10]. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 1-19.
- [11]. James Bennett, Stan Lanning; "The Netflix Prize", In KDD Cup and Workshop in conjunction with KDD, 2007.
- [12]. Mohammad Yahya H. Al-Shamri, Kamal K. Bharadwaj; "A Compact User Model for Hybrid Movie Recommender System" in International Conference on Computational Intelligence and Multimedia Applications 2007
- [13]. Costin-Gabriel Chiru, Vladimir-Nicolae Dinu, Ctina Preda, Matei Macri; "Movie Recommender System Using the User's Psychological Profile" in IEEE International Conference on ICCP, 2015.
- [14]. Christina Christakou, Leonidas Lefakis, Spyros Vrettos and Andreas Stafylopatis; "A Movie Recommender System Based on Semi-Supervised Clustering", IEEE Computer Society Washington, DC, USA 2015.
- [15]. Luis M. de Campos, Juan M. Fernández-Luna *, Juan F. Huete, Miguel A. Rueda-Morales; "Combining

content-based and collaborative recommendations,
revised 2010.